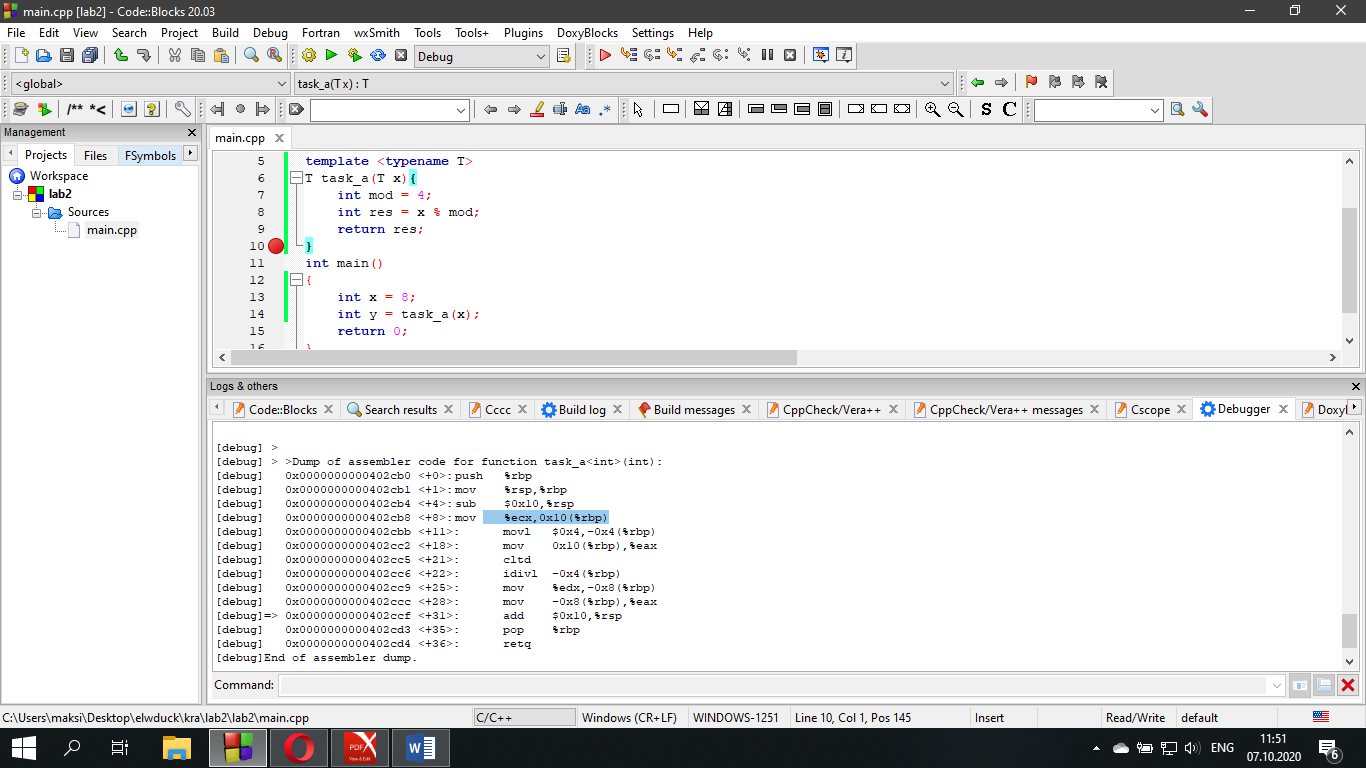
Задание 1



[debug] > >Dump of assembler code for function task\_a<int>(int):

[debug] 0x0000000000402cb0 <+0>: push %rbp

[debug] 0x0000000000402cb1 <+1>: mov %rsp,%rbp

[debug] 0x0000000000402cb4 <+4>: sub $0x10,%rsp

[debug] 0x0000000000402cb8 <+8>: mov %ecx,0x10(%rbp)

[debug] 0x0000000000402cbb <+11>: movl $0x4,-0x4(%rbp)

[debug] 0x0000000000402cc2 <+18>: mov 0x10(%rbp),%eax

[debug] 0x0000000000402cc5 <+21>: cltd

[debug] 0x0000000000402cc6 <+22>: idivl -0x4(%rbp)

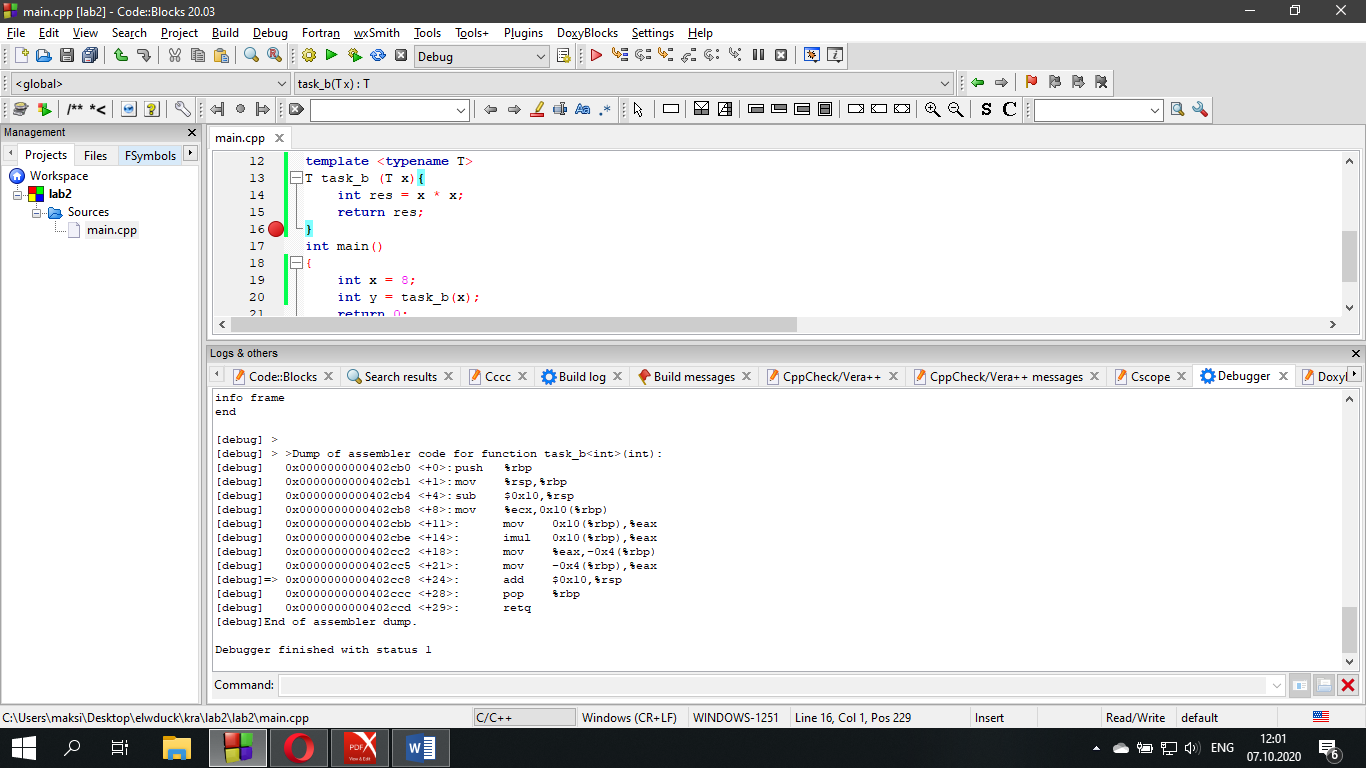
[debug] 0x0000000000402cc9 <+25>: mov %edx,-0x8(%rbp)

[debug] 0x0000000000402ccc <+28>: mov -0x8(%rbp),%eax

[debug]=> 0x0000000000402ccf <+31>: add $0x10,%rsp

[debug] 0x0000000000402cd3 <+35>: pop %rbp

[debug] 0x0000000000402cd4 <+36>: retq



[debug] > >Dump of assembler code for function task\_b<int>(int):

[debug] 0x0000000000402cb0 <+0>: push %rbp

[debug] 0x0000000000402cb1 <+1>: mov %rsp,%rbp

[debug] 0x0000000000402cb4 <+4>: sub $0x10,%rsp

[debug] 0x0000000000402cb8 <+8>: mov %ecx,0x10(%rbp)

[debug] 0x0000000000402cbb <+11>: mov 0x10(%rbp),%eax

[debug] 0x0000000000402cbe <+14>: imul 0x10(%rbp),%eax

[debug] 0x0000000000402cc2 <+18>: mov %eax,-0x4(%rbp)

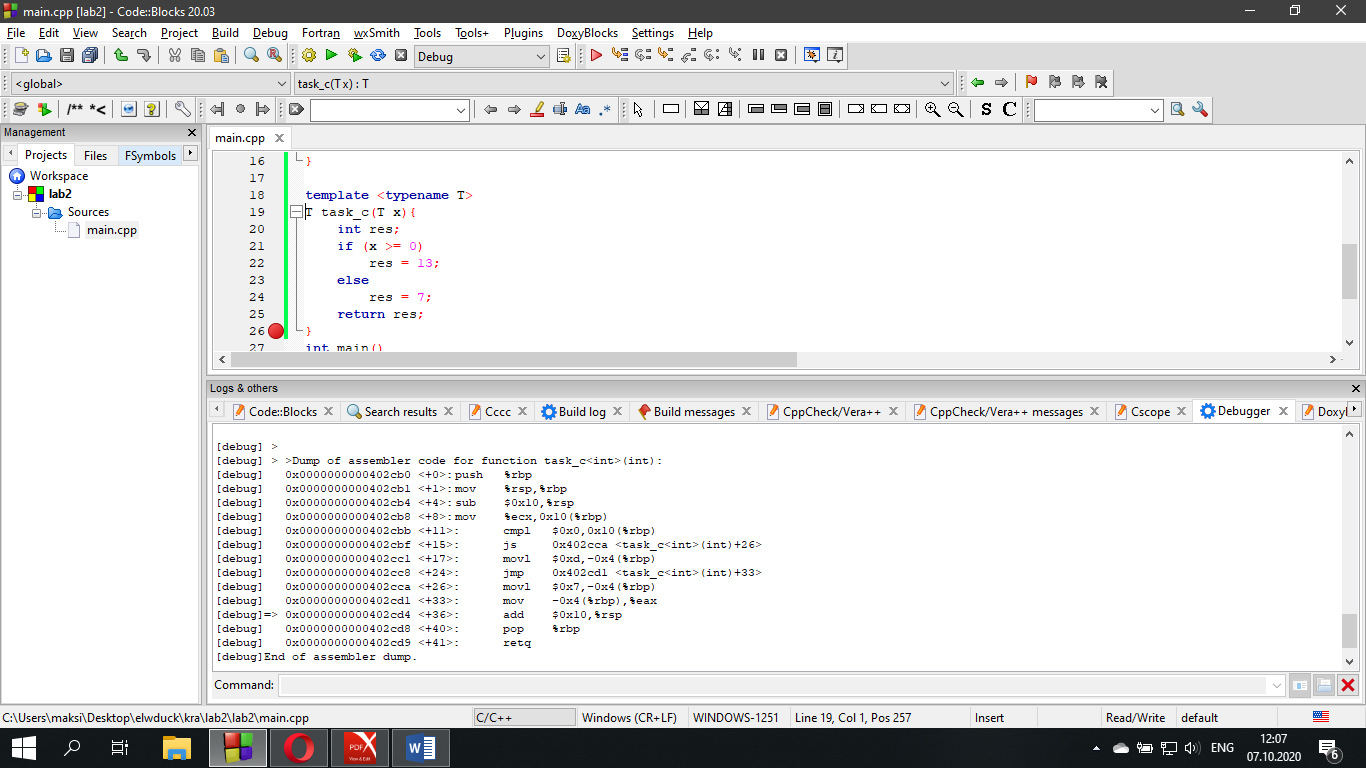
[debug] 0x0000000000402cc5 <+21>: mov -0x4(%rbp),%eax

[debug]=> 0x0000000000402cc8 <+24>: add $0x10,%rsp

[debug] 0x0000000000402ccc <+28>: pop %rbp

[debug] 0x0000000000402ccd <+29>: retq

[debug]End of assembler dump.



[debug] > >Dump of assembler code for function task\_c<int>(int):

[debug] 0x0000000000402cb0 <+0>: push %rbp

[debug] 0x0000000000402cb1 <+1>: mov %rsp,%rbp

[debug] 0x0000000000402cb4 <+4>: sub $0x10,%rsp

[debug] 0x0000000000402cb8 <+8>: mov %ecx,0x10(%rbp)

[debug] 0x0000000000402cbb <+11>: cmpl $0x0,0x10(%rbp)

[debug] 0x0000000000402cbf <+15>: js 0x402cca <task\_c<int>(int)+26>

[debug] 0x0000000000402cc1 <+17>: movl $0xd,-0x4(%rbp)

[debug] 0x0000000000402cc8 <+24>: jmp 0x402cd1 <task\_c<int>(int)+33>

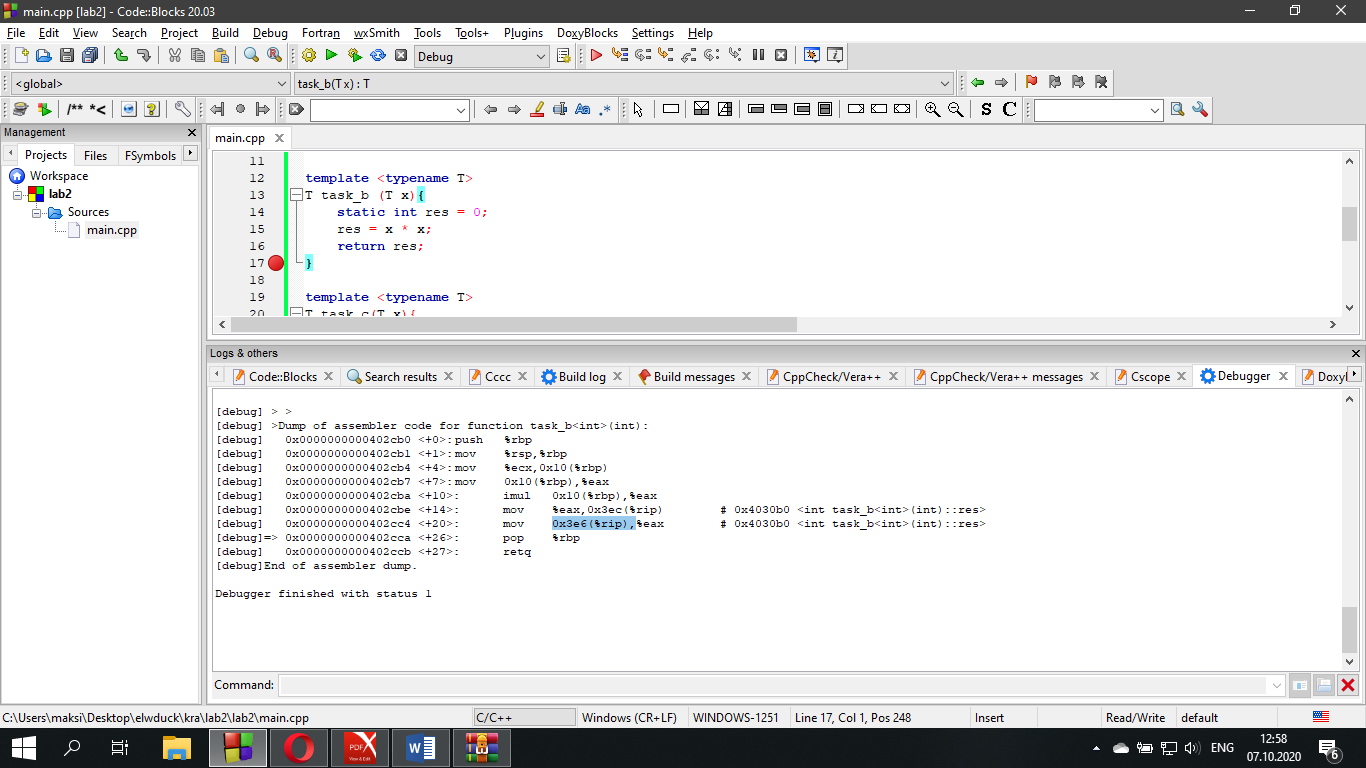
[debug] 0x0000000000402cca <+26>: movl $0x7,-0x4(%rbp)

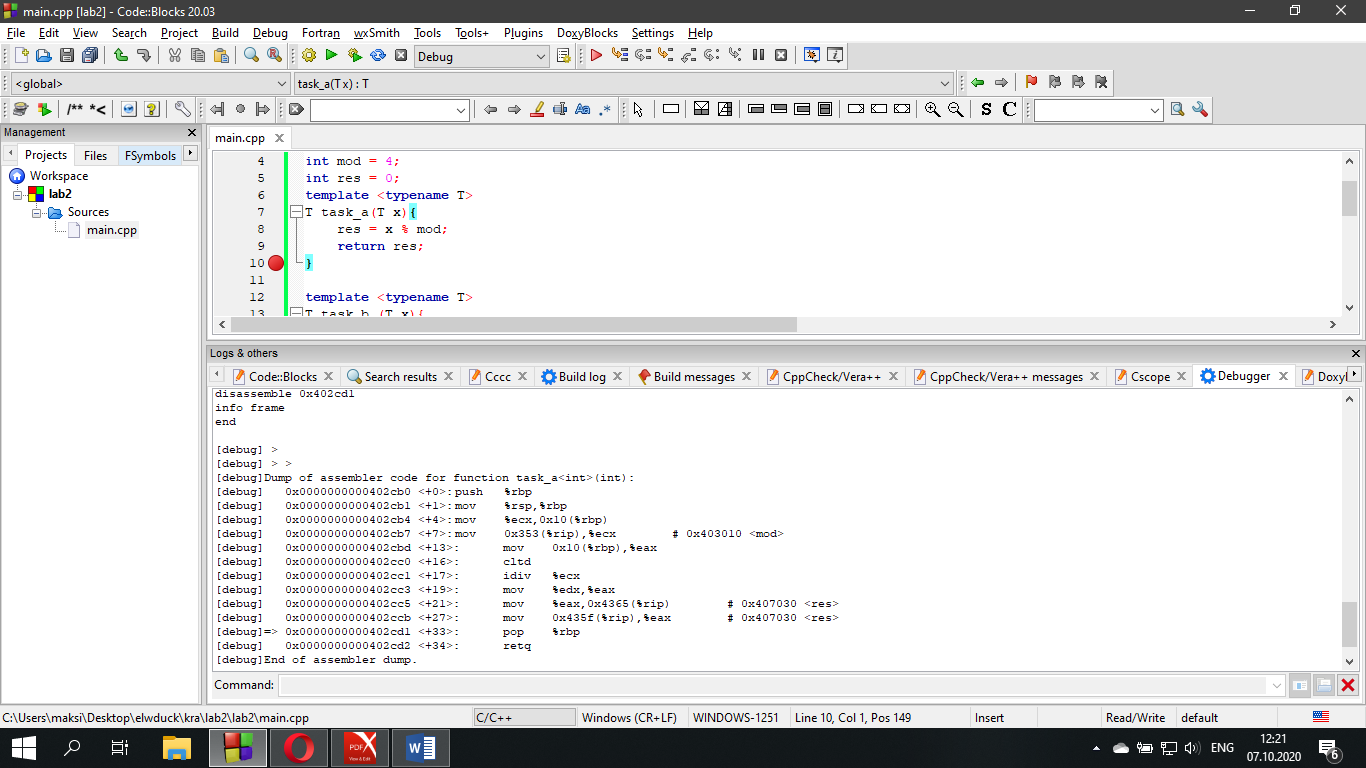
[debug] 0x0000000000402cd1 <+33>: mov -0x4(%rbp),%eax

[debug]=> 0x0000000000402cd4 <+36>: add $0x10,%rsp

[debug] 0x0000000000402cd8 <+40>: pop %rbp

[debug] 0x0000000000402cd9 <+41>: retq





[debug]Dump of assembler code for function task\_a<int>(int):

[debug] 0x0000000000402cb0 <+0>: push %rbp

[debug] 0x0000000000402cb1 <+1>: mov %rsp,%rbp

[debug] 0x0000000000402cb4 <+4>: mov %ecx,0x10(%rbp)

[debug] 0x0000000000402cb7 <+7>: mov 0x353(%rip),%ecx # 0x403010 <mod>

[debug] 0x0000000000402cbd <+13>: mov 0x10(%rbp),%eax

[debug] 0x0000000000402cc0 <+16>: cltd

[debug] 0x0000000000402cc1 <+17>: idiv %ecx

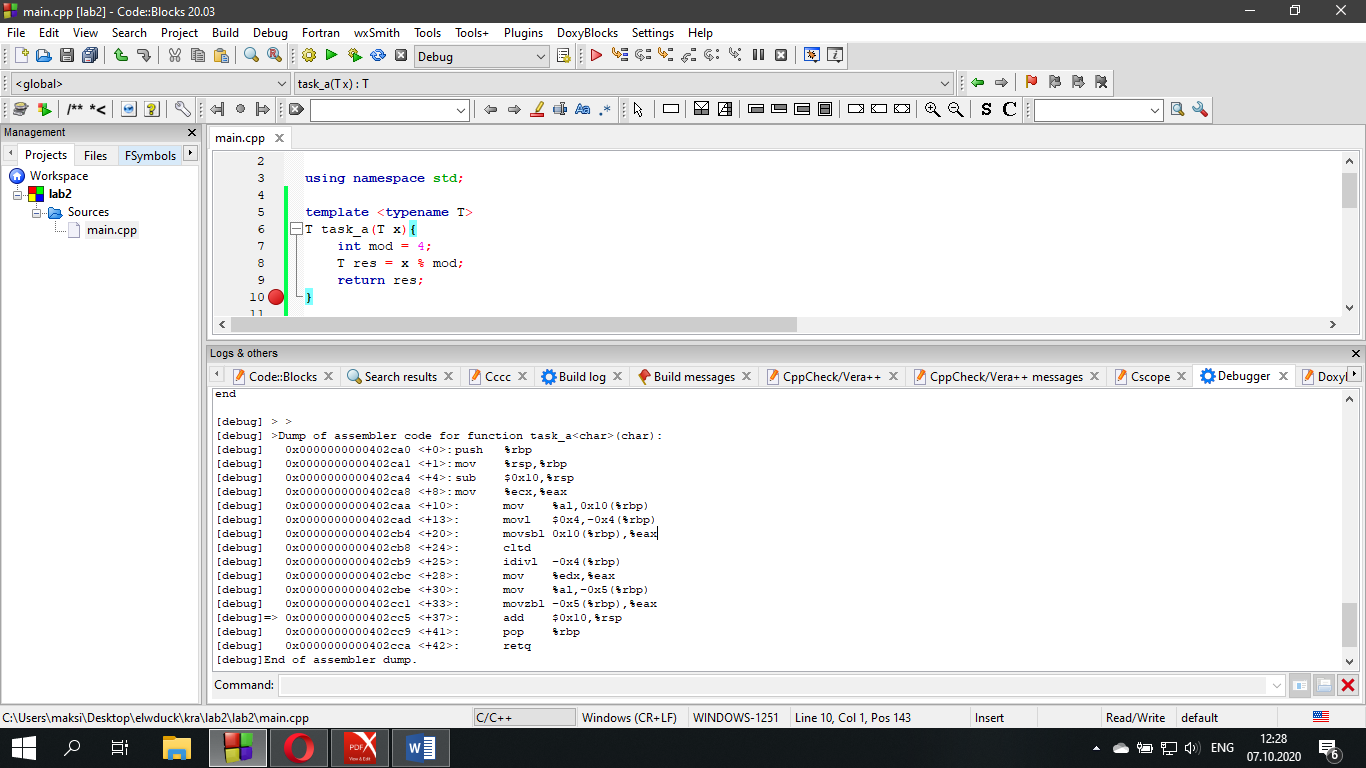
[debug] 0x0000000000402cc3 <+19>: mov %edx,%eax

[debug] 0x0000000000402cc5 <+21>: mov %eax,0x4365(%rip) # 0x407030 <res>

[debug] 0x0000000000402ccb <+27>: mov 0x435f(%rip),%eax # 0x407030 <res>

[debug]=> 0x0000000000402cd1 <+33>: pop %rbp

[debug] 0x0000000000402cd2 <+34>: retq



[debug] >Dump of assembler code for function task\_a<char>(char):

[debug] 0x0000000000402ca0 <+0>: push %rbp

[debug] 0x0000000000402ca1 <+1>: mov %rsp,%rbp

[debug] 0x0000000000402ca4 <+4>: sub $0x10,%rsp

[debug] 0x0000000000402ca8 <+8>: mov %ecx,%eax

[debug] 0x0000000000402caa <+10>: mov %al,0x10(%rbp)

[debug] 0x0000000000402cad <+13>: movl $0x4,-0x4(%rbp)

[debug] 0x0000000000402cb4 <+20>: movsbl 0x10(%rbp),%eax

[debug] 0x0000000000402cb8 <+24>: cltd

[debug] 0x0000000000402cb9 <+25>: idivl -0x4(%rbp)

[debug] 0x0000000000402cbc <+28>: mov %edx,%eax

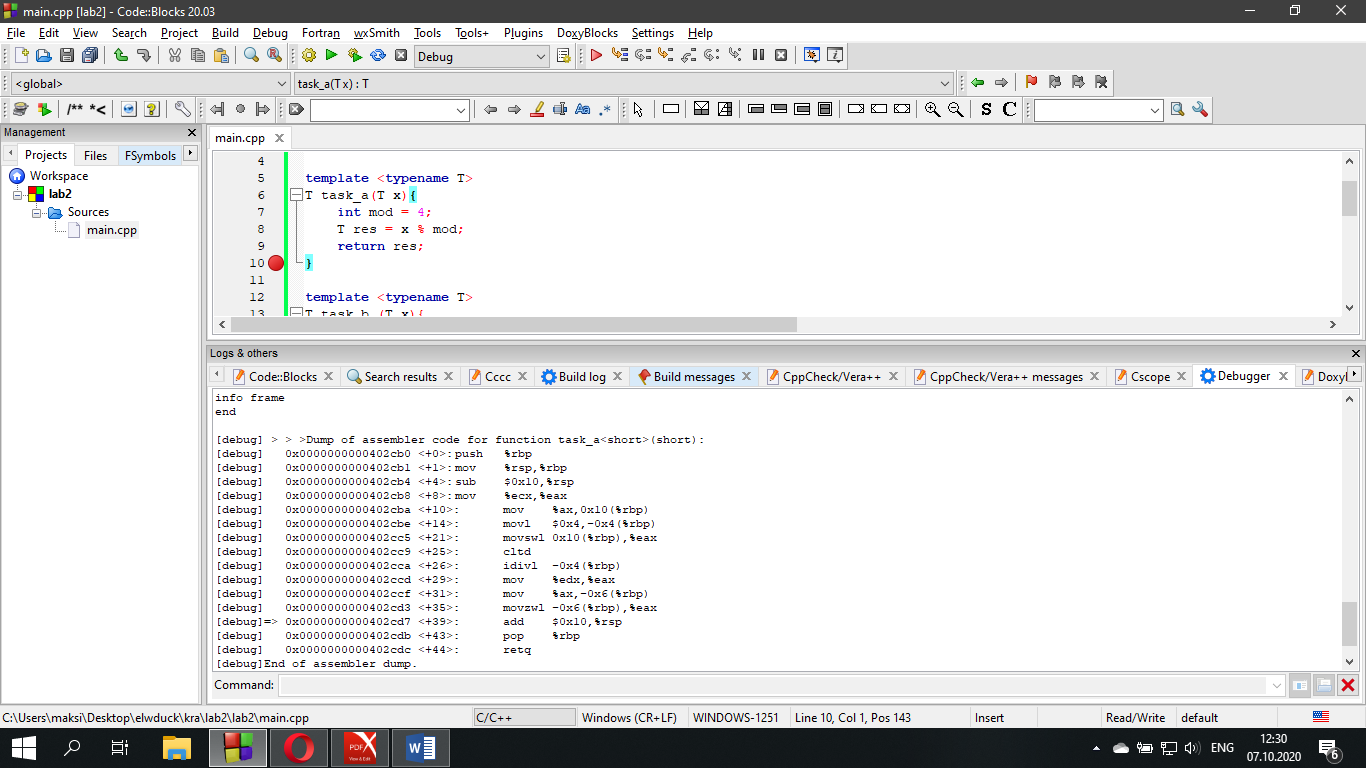
[debug] 0x0000000000402cbe <+30>: mov %al,-0x5(%rbp)

[debug] 0x0000000000402cc1 <+33>: movzbl -0x5(%rbp),%eax

[debug]=> 0x0000000000402cc5 <+37>: add $0x10,%rsp

[debug] 0x0000000000402cc9 <+41>: pop %rbp

[debug] 0x0000000000402cca <+42>: retq

[debug] > > >Dump of assembler code for function task\_a<short>(short):

[debug] 0x0000000000402cb0 <+0>: push %rbp

[debug] 0x0000000000402cb1 <+1>: mov %rsp,%rbp

[debug] 0x0000000000402cb4 <+4>: sub $0x10,%rsp

[debug] 0x0000000000402cb8 <+8>: mov %ecx,%eax

[debug] 0x0000000000402cba <+10>: mov %ax,0x10(%rbp)

[debug] 0x0000000000402cbe <+14>: movl $0x4,-0x4(%rbp)

[debug] 0x0000000000402cc5 <+21>: movswl 0x10(%rbp),%eax

[debug] 0x0000000000402cc9 <+25>: cltd

[debug] 0x0000000000402cca <+26>: idivl -0x4(%rbp)

[debug] 0x0000000000402ccd <+29>: mov %edx,%eax

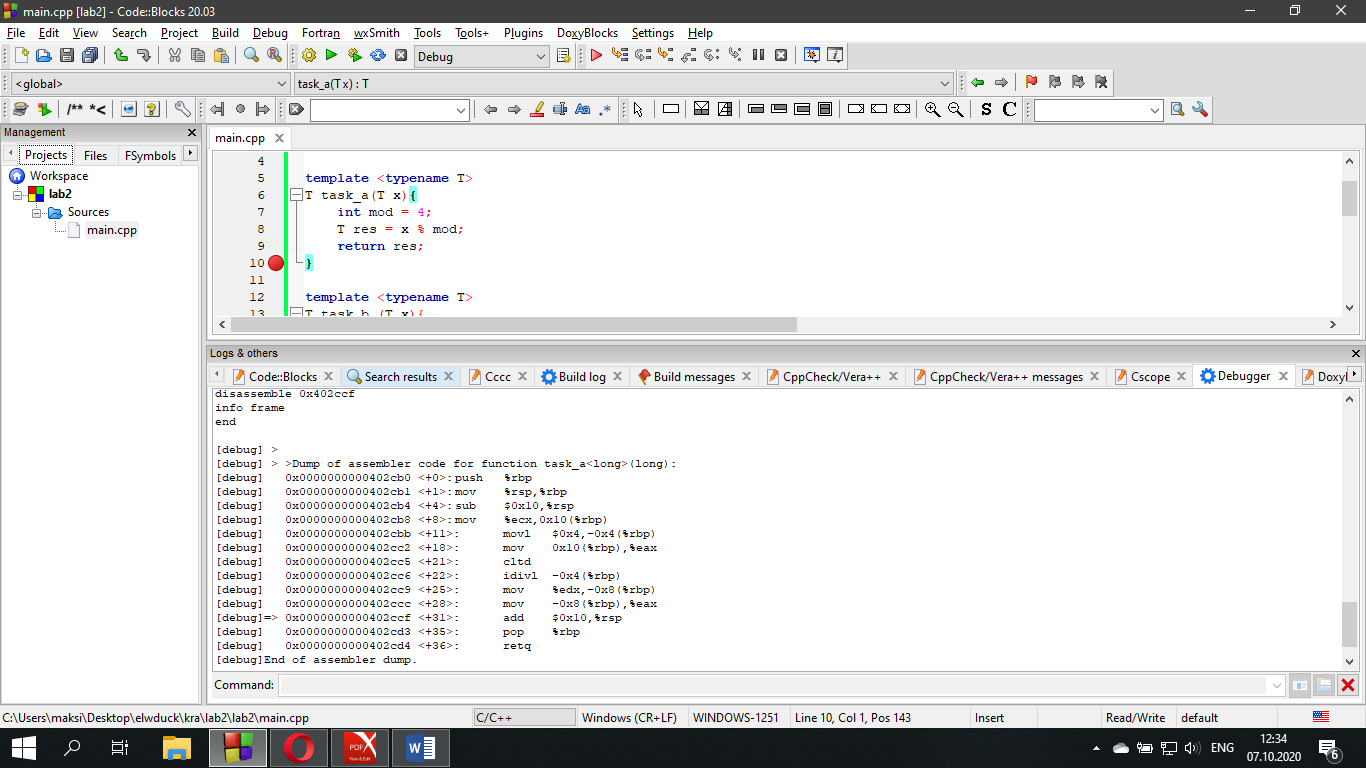
[debug] 0x0000000000402ccf <+31>: mov %ax,-0x6(%rbp)

[debug] 0x0000000000402cd3 <+35>: movzwl -0x6(%rbp),%eax

[debug]=> 0x0000000000402cd7 <+39>: add $0x10,%rsp

[debug] 0x0000000000402cdb <+43>: pop %rbp

[debug] 0x0000000000402cdc <+44>: retq

[debug] > >Dump of assembler code for function task\_a<long>(long):

[debug] 0x0000000000402cb0 <+0>: push %rbp

[debug] 0x0000000000402cb1 <+1>: mov %rsp,%rbp

[debug] 0x0000000000402cb4 <+4>: sub $0x10,%rsp

[debug] 0x0000000000402cb8 <+8>: mov %ecx,0x10(%rbp)

[debug] 0x0000000000402cbb <+11>: movl $0x4,-0x4(%rbp)

[debug] 0x0000000000402cc2 <+18>: mov 0x10(%rbp),%eax

[debug] 0x0000000000402cc5 <+21>: cltd

[debug] 0x0000000000402cc6 <+22>: idivl -0x4(%rbp)

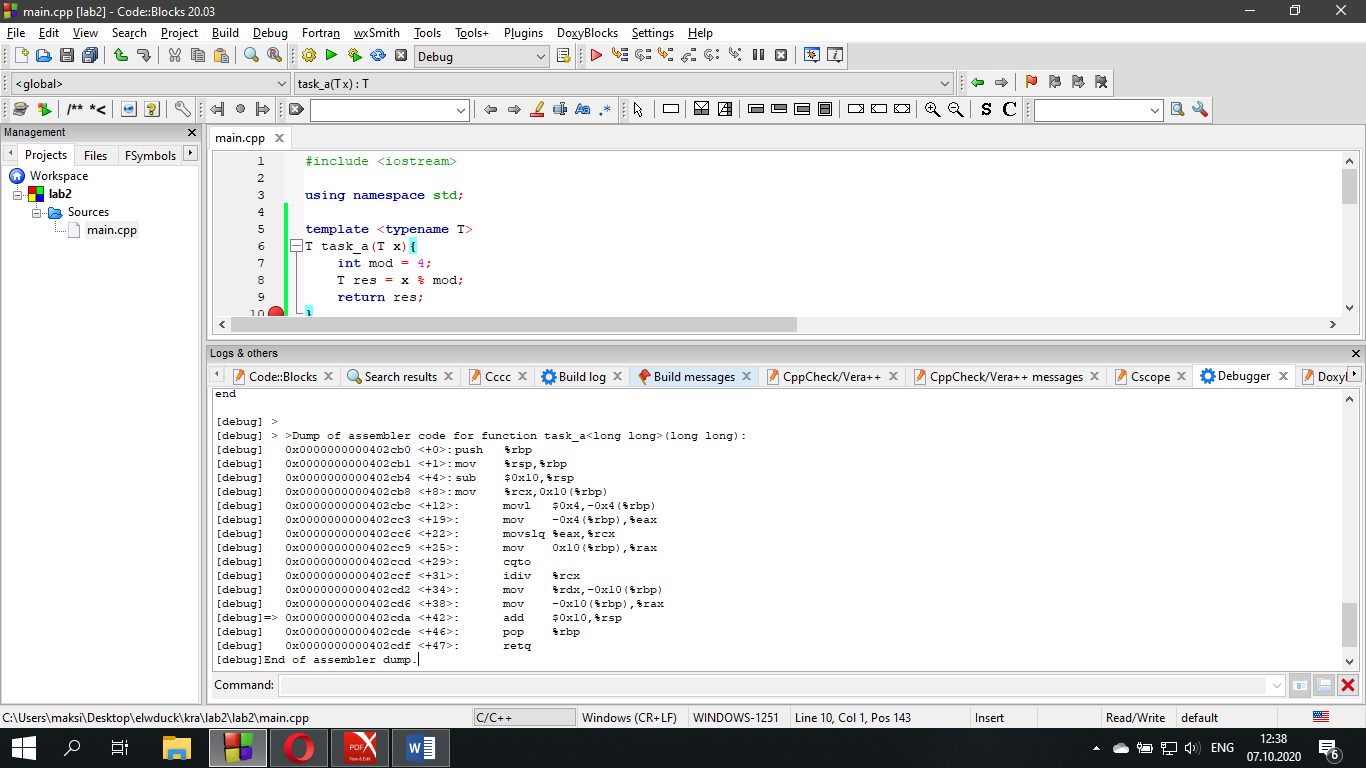
[debug] 0x0000000000402cc9 <+25>: mov %edx,-0x8(%rbp)

[debug] 0x0000000000402ccc <+28>: mov -0x8(%rbp),%eax

[debug]=> 0x0000000000402ccf <+31>: add $0x10,%rsp

[debug] 0x0000000000402cd3 <+35>: pop %rbp

[debug] 0x0000000000402cd4 <+36>: retq

[debug] > >Dump of assembler code for function task\_a<long long>(long long):

[debug] 0x0000000000402cb0 <+0>: push %rbp

[debug] 0x0000000000402cb1 <+1>: mov %rsp,%rbp

[debug] 0x0000000000402cb4 <+4>: sub $0x10,%rsp

[debug] 0x0000000000402cb8 <+8>: mov %rcx,0x10(%rbp)

[debug] 0x0000000000402cbc <+12>: movl $0x4,-0x4(%rbp)

[debug] 0x0000000000402cc3 <+19>: mov -0x4(%rbp),%eax

[debug] 0x0000000000402cc6 <+22>: movslq %eax,%rcx

[debug] 0x0000000000402cc9 <+25>: mov 0x10(%rbp),%rax

[debug] 0x0000000000402ccd <+29>: cqto

[debug] 0x0000000000402ccf <+31>: idiv %rcx

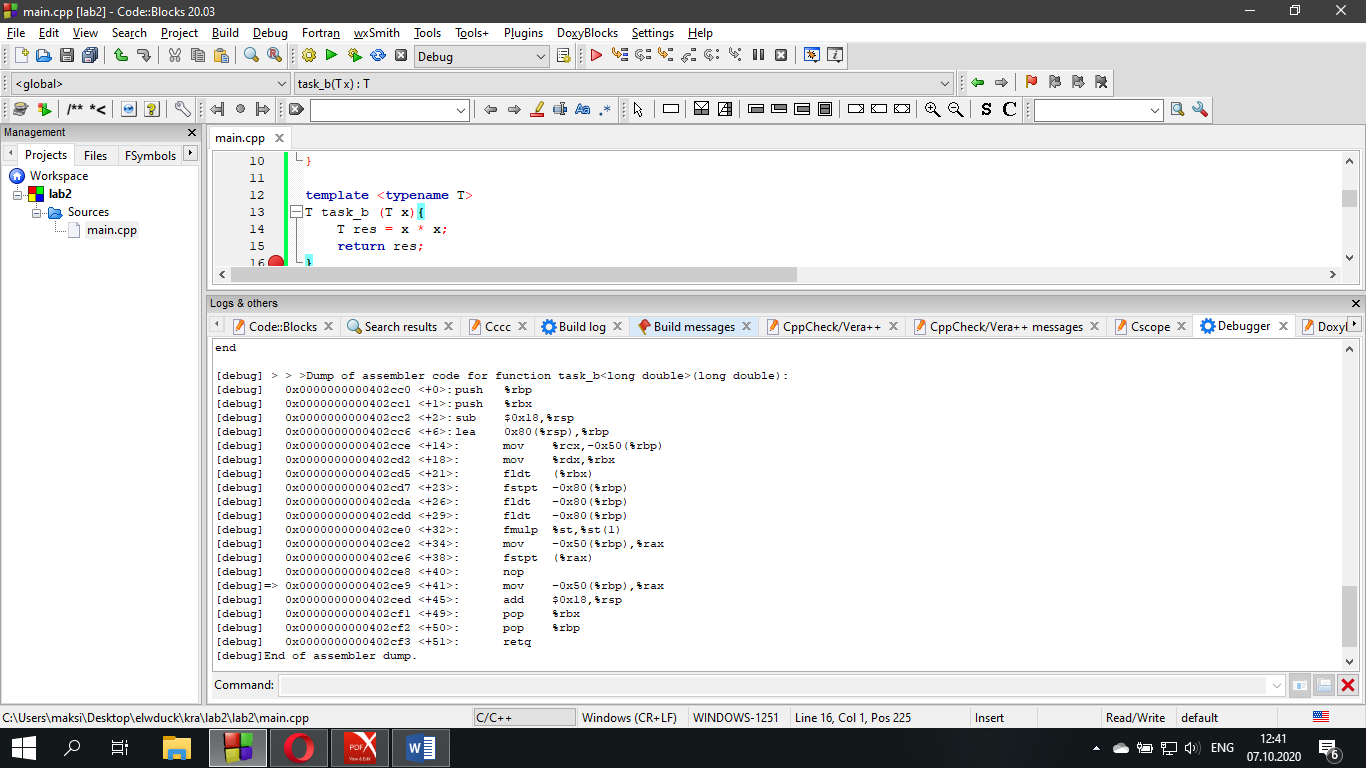
[debug] 0x0000000000402cd2 <+34>: mov %rdx,-0x10(%rbp)

[debug] 0x0000000000402cd6 <+38>: mov -0x10(%rbp),%rax

[debug]=> 0x0000000000402cda <+42>: add $0x10,%rsp

[debug] 0x0000000000402cde <+46>: pop %rbp

[debug] 0x0000000000402cdf <+47>: retq

[debug] > > >Dump of assembler code for function task\_b<long double>(long double):

[debug] 0x0000000000402cc0 <+0>: push %rbp

[debug] 0x0000000000402cc1 <+1>: push %rbx

[debug] 0x0000000000402cc2 <+2>: sub $0x18,%rsp

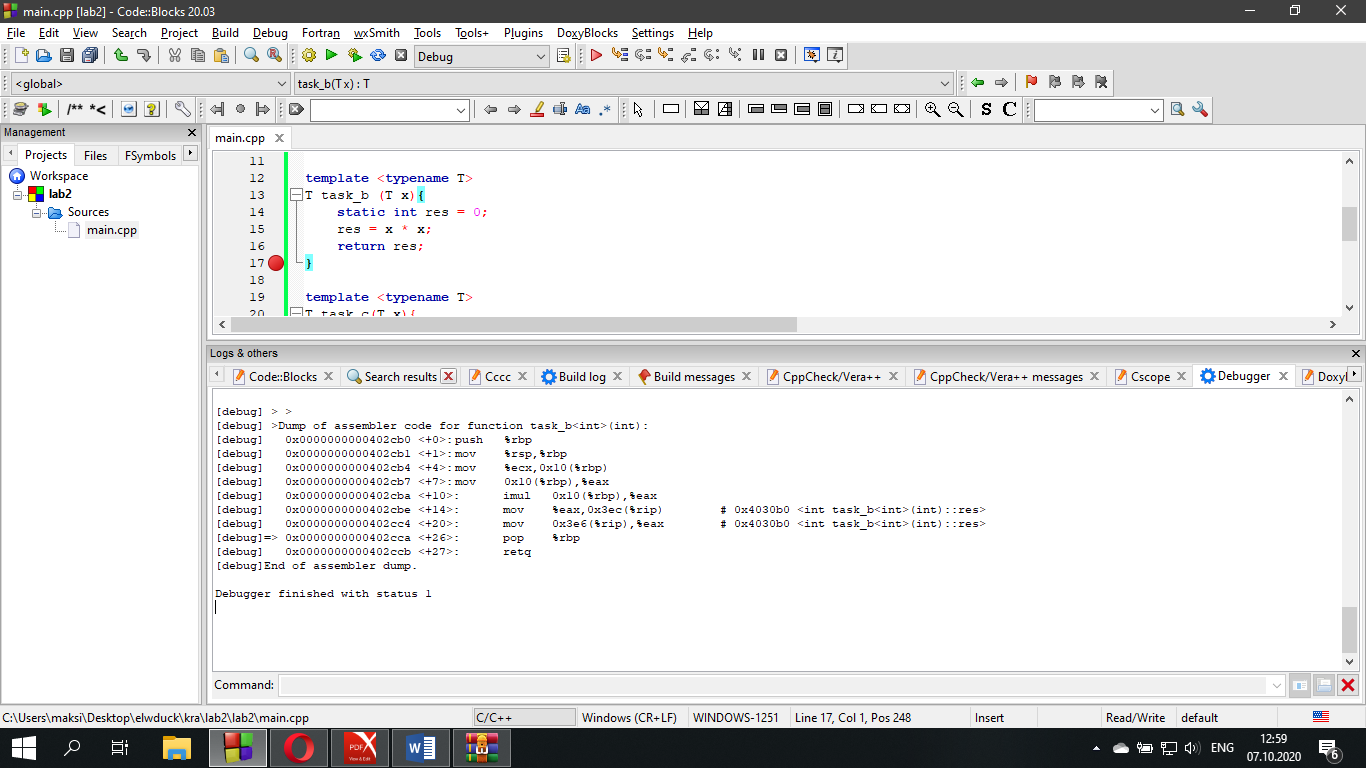
[debug] 0x0000000000402cc6 <+6>: lea 0x80(%rsp),%rbp //эффективный адрес источника

[debug] 0x0000000000402cce <+14>: mov %rcx,-0x50(%rbp)

[debug] 0x0000000000402cd2 <+18>: mov %rdx,%rbx

[debug] 0x0000000000402cd5 <+21>: fldt (%rbx)

[debug] 0x0000000000402cd7 <+23>: fstpt -0x80(%rbp) //сохранить вещественное значение с извлечением из стека

[debug] >Dump of assembler code for function task\_b<int>(int):

[debug] 0x0000000000402cb0 <+0>: push %rbp

[debug] 0x0000000000402cb1 <+1>: mov %rsp,%rbp

[debug] 0x0000000000402cb4 <+4>: mov %ecx,0x10(%rbp)

[debug] 0x0000000000402cb7 <+7>: mov 0x10(%rbp),%eax

[debug] 0x0000000000402cba <+10>: imul 0x10(%rbp),%eax

[debug] 0x0000000000402cbe <+14>: mov %eax,0x3ec(%rip) # 0x4030b0 <int task\_b<int>(int)::res>

[debug] 0x0000000000402cc4 <+20>: mov 0x3e6(%rip),%eax # 0x4030b0 <int task\_b<int>(int)::res>

[debug]=> 0x0000000000402cca <+26>: pop %rbp

[debug] 0x0000000000402ccb <+27>: retq